

Team 04: Trainingstracker

Egorow Timothy
Schifferl Kilian
Al Ghazzawi Badee
Friedrich Jeremias
Evangelou Katerina

1. Änderungsgeschichte

Datum	Version	Änderung	Autor
25.03.22	1.0	Erstellung	Egorow
24.04.22	2.0	Workpackages	Egorow
30.06.22	3.0	Fehler	Egorow

2. Einführung

2.1 Referenzen

- Craig Larman: UML 2 und Patterns angewendet
- Vorlesungsmaterialien der Software-Engineering Vorlesung

3. Projekt Übersicht

Das Projekt soll Menschen dabei unterstützen ihr Training zu tracken, sowie die Körpermaße und Kalorien.

Dem Benutzer werden folgende Optionen zur Verfügung gestellt:

- Das Erstellen eines Trainingsplans
- Das Tracken des Trainings
- Das Zählen von Kalorien
- Das Speichern von Körpermaßen

3.1 Zweck und Ziele

Dieses Projekt soll Menschen dabei unterstützen ihr Training verfolgen zu können. Die Verfolgung des Fortschrittes ist wichtig um sich zu steigern, als auch motiviert zu bleiben.

3.2 Annahmen und Einschränkungen

Es wird eine Datenbank benötigt, auf welcher wir die Daten speichern können. Diese ist essentiell wichtig, daher muss die Funktionalität gewährleistet sein.

Wir nutzen eine REST-API. Mit dieser haben nur wenige Personen aus unserer Gruppe Erfahrung. Daher muss jeder Person, welche im Backend-Bereich tätig sein will, sich einarbeiten.

Jede Person ist dafür verantwortlich, dass die eigene Zeit grob erfasst wird, um später das Zusammenführen der Daten zu erleichtern.

Die Dokumente werden während des Projektes aktualisiert und verfeinert.

3.3 Ergebnisse

- Projektplan
- Anforderungen
- Architecture
- Backenddokumentation
- Build-fähiges Programm

4. Projekt Organisation

Das voraussichtlich schwerste Paket ist die Datenbank und der REST-API-Server. Darin muss etwas Zeit investiert werden, um eine gute, funktionierende Schnittstelle zu schaffen.

Als nächstes werden die Funktionalitäten behandelt, diese sollten keine allzu große Herausforderung darstellen. Parallel dazu wird die GUI entworfen.

4.1 Organisation's Struktur

Rolle	Verantwortlicher
Admin	Friedrich
Webseite	Evangelou
Dokumentation	Egorow
Test	Schifferl
Build	Schifferl
Front-end	Evangelou
Back-end	Friedrich

4.2 Externe Schnittstellen

- Datenbank
- API-Server

5. Management Prozess

5.1 Projekt Kostenvoranschlag

Für das Projekt stehen 16 Wochen Zeit zur Verfügung. Pro Person werden 150 Stunden benötigt, daraus folgt, dass jede Woche ca. 9 Stunden pro Person aufgebracht werden sollten.

5.2 Projekt Plan

5.2.1 Phase Plan

Meilenstein 1:

- Projektplan
- Anforderungen
- Domainmodell

Meilenstein 2:

- Ein Trainingsplan kann erstellt und gespeichert werden
- Ein Trainingsplan wird gespeichert
- Ein User kann sich einloggen
- Ein überarbeiteter Projektplan liegt vor
- Eine Architektur liegt vor

Meilenstein 3:

- Der Kalorienzähler funktioniert
- Die Körpermaße können erfasst werden
- Die UI wurde verbessert
- Die Dokumentation ist überarbeitet worden

Meilenstein 4:

- Die letzten Fehler wurden behoben
- Das Programm ist buildfähig
- Die Dokumentation ist vollständig

5.2.2 Meeting

Es ist ein wöchentliches Meeting am Dienstag von 19:00-20:00 beschlossen worden. In diesem wird über den bisherigen Fortschritt diskutiert und Fragen des einzelnen geklärt. Dieses Meeting findet als Videokonferenz statt.

5.2.3 Release

Typ	Datum
Prototyp	19.05.22
Endversion	08.07

6. Risk Management

- Die Datenbank funktioniert nicht wie gewünscht → sofortiges Überarbeiten, Funktionalität sicherstellen
- Der API-Code ist falsch → Besprechen mit Jeremias
- Die Test funktionieren nicht wie gewünscht → überarbeiten der Planung der Tests
- Der Code funktioniert nicht wie gewünscht → Fehler suchen, beheben und Dokumentieren

7. Arbeitspakete

Back-end

Datenbank	Jerry	125H
Server	Jerry	125H
User-Controllers	Killian	40H
Workoutroutine-Controllers	Timothy	40H
Product-Controllers	Timothy	40H
Exercise-Controllers	Timothy	40H

Front-end

Login-Fenster	Katharina	10H
User-Fenster	Killian	15H
Workoutroutine-Fenster	Katherina	10H
Caloriecounter-Fenster	Katherina	10H
Exercise-Fenster	Katherina	10H
UserParameters-Fenster	Killian	15H

- Datenbank 125H
- Server 125H
- Ui 70H
- Modell 100H
- Controller 100H
- Doks 100H

8. Infrastruktur

- Clion
- QtCreator
- Visual paradigm
- Docker
- PostgreSQL
- GitLab
- Insomnia
- DataGrip
-

9. Unterstützende Prozesse

Konfiguration:

Für die Konfiguration wird GitLab verwendet. Darin werden unsere Codeteile gespeichert, und nach dem Testen zusammengefügt. Für das Testen ist Herr Egorow zuständig, für das Zusammenbauen Herr

Schifferl. Die Zustände werden abgespeichert und sind damit wiederherstellbar.

Testen:

Zunächst werden einfache Smoke-tests durchgeführt, um grobe Fehler zu finden. Sobald die Software umfangreicher wird, werden Unit-tests genutzt um automatisiert zu testen.

Zeiterfassung:

Für die Zeiterfassung der Gruppe ist Herr Egorow zuständig. Jede Person erfasst seine eigene Zeit. Bei jedem Meeting wird diese dann durch Herrn Egorow in einer Excel Tabelle erfasst.